

Building an Ubuntu Virtual Machine for MATLAB

Tristram Scott *

August 1, 2011

1 Introduction

MATLAB is no longer offered on Solaris platforms, and has never been offered for Solaris x86, only SPARC. For those who wish to use MATLAB under Unix, the current choices are Linux or MacOS X. This document describes one way of providing a Linux *virtual machine* for the sole purpose of running MATLAB. The example is for an Ubuntu server *guest* (virtual machine) running on top of a Solaris *host* machine.

2 Document conventions

Different typefaces are used to indicate filenames, commands to be typed etc:

File or directory name	<code>/some/path/somefile.txt</code>
Comand to be typed	command
Command output	response
Normal user's shell prompt	tcsh%
Root user's shell prompt	tcsh#
Environment variables	VARIABLE

Long lines have been wrapped to fit within the typed page. For commands, continuation is indicated using the backslash (\). For command output, the wrapped lines are indented. Command output is usually not shown, unless it is relevant for the subsequent text.

*Id: BuildVM.tex,v 1.2 2010/08/17 13:30:22 tristram Exp tristram

3 *VirtualBox*

There are now many virtual machines available, each with their own benefits. For this document, we have used *VirtualBox* ¹, version 3.2.6. It is simple enough to install following the examples in the user manual. *VirtualBox* is offered for a number of host platforms, including:

- Solaris
- Windows
- MacOS X
- Linux

The range of guest platforms known to work with *VirtualBox* covers all of the above, plus a number of others. Virtual machines can be cloned, migrated from one host to another, even while running, and it is a simple matter to alter the offered hardware, e.g. adding extra RAM or CPUs.

In what follows I assume you have already installed *VirtualBox*, so if that is not the case you will need to do so before proceeding.

4 **Ubuntu server**

There are many Linux platforms available, and the process of choosing amongst them is somewhat daunting. I have used CentOS for several years, but never been entirely happy with its performance or stability as a virtual machine². I recently went looking for an alternative Linux distribution, and have found Ubuntu³ meets my needs:

- Simple installation.
- Fast boot times.
- Small footprint.
- Runs latest version of MATLAB.
- Available as a server rather than desktop version.

¹<http://www.virtualbox.org/>

²I am not meaning to suggest that CentOS is not a fine and stable operating system, just observing that my experience of running it under *VirtualBox* has not been good. In particular, I have never managed to achieve a reliable system clock.

³<http://www.ubuntu.com/server>

- Installs from a single CD.

Whatever your choice of Linux, you will need either to have an installation CD / DVD at hand, or to download an ISO image of the installation media. For this example I assume you have downloaded a copy of the install image and saved it locally somewhere on the host machine.

5 Creating the virtual machine

If you have a method of running the graphical interface, I recommend using the GUI, and creating the new virtual machine from there. Assuming you have *VirtualBox* installed on your system, it should start with the following command:

```
tcsh% VirtualBox &
```

The ampersand runs the command in the background, returning you to the command prompt straight away. Follow the examples in the *VirtualBox* user manual to create the guest.

Alternatively, the virtual machine, and its associated virtual disk, can be created directly from the command line. This is useful where the host doesn't have X installed, for example, or just when it is at the far end of a low bandwidth connection:

```
tcsh% VBoxManage createvm --name ubuntu --register \
--ostype Ubuntu_64
Oracle VM VirtualBox Command Line Management Interface
Version 3.2.6
(C) 2005-2010 Oracle Corporation
All rights reserved.

Virtual machine 'ubuntu' is created and registered.
UUID: 7aabdc0-269b-4ce8-a11e-f59ac12b1945
Settings file:
'/home/tristram/.VirtualBox/Machines/ubuntu/ubuntu.xml'
tcsh% VBoxManage modifyvm ubuntu --memory 4096 \
--cpus 2 --nic1 bridged --boot1 dvd \
--bridgeadapter1 "e1000g0 - Intel PRO/1000 Gigabit Ethernet"
```

The name used for the **bridgeadapter1** is host specific. To find what *VirtualBox* calls your physical network interfaces, use the **VBoxManage**

list hostifs command:

```
tcsh% VBoxManage list hostifs
```

```
Oracle VM VirtualBox Command Line Management Interface  
Version 3.2.6  
(C) 2005-2010 Oracle Corporation  
All rights reserved.
```

```
Name: e1000g0 - Intel PRO/1000 Gigabit Ethernet  
GUID: 30303165-6730-4030-8000-00144f4aa46f  
Dhcp: Disabled  
IPAddress: 192.168.0.2  
NetworkMask: 255.255.255.0  
IPV6Address:  
IPV6NetworkMaskPrefixLength: 0  
HardwareAddress: 00:14:4f:4a:a4:6f  
MediumType: Ethernet  
Status: Up  
VBoxNetworkName: HostInterfaceNetworking-e1000g0 - Intel  
PRO/1000 Gigabit Ethernet
```

It may be useful to create more than one network interface for the virtual machine. If I am wanting to migrate to a new virtual machine I sometimes find it useful to keep the same MAC address for the primary ethernet interface. The reason for this is that the MAC address is often the thing which software licenses are keyed to. If we use the same MAC address on the new machine, the existing license keys will work without having to go through the steps of software registration and activation. Of course, networking is not going to function correctly if two virtual machines are running at the same time, using the same MAC address. To avoid such issues, ensure that only one of the virtual machines is alive at a time⁴.

To add an additional network interface, and to bind a specific MAC address to it, use something like the following:

```
tcsh% VBoxManage modifyvm ubuntu --nic2 bridged \  
--macaddress2 08001234ABCD \  
--bridgeadapter2 "e1000g0 - Intel PRO/1000 Gigabit Ethernet"
```

If the intention is to control the MAC address for the primary interface,

⁴Alternatively, configure the network interfaces so they will never talk to each other, nor to the same third party machine. This can usually be achieved by keeping them on separate private subnets which are not utilised by any other machines.

then it can be altered as follows:

```
tcsh% VBoxManage modifyvm ubuntu --macaddress1 08001234CDEF
```

5.1 Create a new virtual hard disk

Now we have a new guest machine, but as yet it has no storage. We will create a new virtual hard disk image, add a SATA controller to the guest, and set the new disk as the first virtual disk on the guest:

```
tcsh% VBoxManage createhd --filename ubuntu-disk.vdi \  
--size 10240 --variant Fixed  
tcsh% VBoxManage storagectl ubuntu --name StorageController \  
--add sata  
tcsh% VBoxManage storageattach ubuntu --storagectl StorageController \  
--port 0 --device 0 --type hdd --medium ubuntu-disk.vdi
```

For our purpose, creating a guest machine to run MATLAB, 10 GB of disk space is plenty. The intention is only have the operating system and the MATLAB installation stored on this disk. Home directories etc. will be automounted via nfs from another system. The **--variant Fixed** option creates the full hard disk image immediately, rather than growing it as space is required. This can make quite a difference to initial performance, especially when nfs mounting the directory where the disk image is stored.

We also need to register the ISO file that contains the installation media, and attach the dvd to the IDE controller:

```
tcsh% VBoxManage storagectl ubuntu --name StorageController2 \  
--add ide  
tcsh% VBoxManage storageattach ubuntu --storagectl StorageController2 \  
--port 0 --device 1 --type dvddrive --medium \  
/rpool/tmp/ubuntu-10.04-server-amd64.iso
```

5.2 Start the virtual machine

It should now be possible to start the guest, and connect to it using `rdesktop`:

```
tcsh% VBoxHeadless -s ubuntu &  
tcsh% rdesktop localhost:3389
```

If all has gone well, the remote desktop should display the *VirtualBox* splash screen as it boots the guest, and then shortly afterwards you will

see the first screen of the Ubuntu installation, prompting you to choose a language. Assuming this has worked, proceed to Section 6.

5.3 Destroy the virtual machine

If you need to completely remove the guest machine, the following commands should do the trick:

```
tcsh% VBoxManage storagectl ubuntu --name StorageController --remove
tcsh% VBoxManage storagectl ubuntu --name StorageController2 --remove
tcsh% VBoxManage unregistervm ubuntu --delete
tcsh% VBoxManage closemedium disk ubuntu-disk.vdi --delete
tcsh% VBoxManage closemedium dvd \
    /rpool/tmp/ubuntu-10.04-server-amd64.iso
```

Note: The `--delete` option removes all traces of the guest and its disk from the host. Note also that we refrain from removing the downloaded `ubuntu-10.04-server-amd64.iso` image.

6 Installing Ubuntu onto the guest machine

The first screen on the Ubuntu installation is for choosing the language. Pick your favourite and continue.

The next screen gives you options to:

- Install Ubuntu Server
- Install Ubuntu Enterprise Cloud
- Check disk for defects
- Test memory
- Boot from first hard disk
- Rescue a broken system

Choose the first option, *Install Ubuntu server*. After a few moments the installation configuration begins. You need to do the following (my suggested answers in brackets):

- Choose a language (**English**)
- Choose a country (**United Kingdom**)

- Detect keyboard layout (**No**)
- Origin of the keyboard (**United Kingdom**)
- Keyboard layout (**United Kingdom**)

The installation process does a bit more work then returns with more questions. Obviously you should use your own appropriate values, especially for the network configuration.

- Name server address (**192.168.0.6**)
- Hostname (**frog**)
- Partitioning method (**Guided - use entire disk**)
- Select disk to partition (**SCSI3 (0,0,0) (sda) - 10.7 GB ATA VBOX HARDDISK**)
- Partition disks. Write changes (**Yes**)

This last step writes the new partition information to the chosen disk, which for us is the virtual disk image rather than any physical disk from the host system. The process then continues under the heading of *Installing the base system* which may take a couple of minutes. The next section prompts you for user information:

- Full name for the new user (**Tristram Scott**)
- Username for your account (**tristram**)
- Choose a password for the new user (*********)
- Re-enter password to verify (*********)
- Encrypt your home directory (**No**)
- HTTP proxy information (**)**
- How do you want to manage upgrades on this system (**No automatic updates**)

The next screen prompts you for additional packages to install. I choose none at this stage, and then add the required packages manually once the installation is complete. The next heading is *Select and install software*, and once you set it on its way this will again take a few minutes. You will next

be prompted to install the GRUB boot loader (to the virtual disk image), which you should do.

Finally, you are prompted to boot into the new system, remembering to first eject the CD⁵. Press return and it will quickly reboot and present you with a login prompt. Login using the username you specified earlier, and then shutdown the system so we can be sure the DVD is no longer attached:

```
tristram@frog:~$ sudo shutdown -P -t 0 now
```

```
[sudo] password for tristram:
```

Once it has shut down, on the host system, detach the DVD:

```
tcsh% VBoxManage storageattach ubuntu --storagectl StorageController2 \  
--port 0 --device 1 --type dvddrive --medium none
```

7 Installing additional packages on the Ubuntu server

The default installation of Ubuntu Server is very minimal, and so there are a few extra packages required before MATLAB will work properly. To begin with, I usually modify the network settings to use a static IP address, rather than the default DHCP. The file to modify is `/etc/network/interfaces`. We need to change the last line to use static rather than dhcp, and also to add a line each for the address, netmask, and gateway. Edit the file using the following command:

```
tristram@frog:~$ sudo vi /etc/network/interfaces
```

The entire file will look something like this:

⁵It seems that removing the CD is not actually required at this stage. If you are presented with the language choice, choose English, then boot from the first hard disk.


```
# This file describes the network interfaces available on
    your system
# and how to activate them.  For more information, see
    interfaces(5).

# The loopback network interface
auto lo
iface lo inet loopback

# The primary network interface
auto eth0
iface eth0 inet static
address 192.168.0.29
netmask 255.255.255.0
gateway 192.168.0.1
```

The eth0 name for the network interface comes from the (automatically created) definition in `/etc/udev/rules.d/70-persistent-net.rules`. If the attributes of the virtual network interfaces are ever modified after the initial installation of Ubuntu, new definitions will appear in this file (after reboot). This includes modification of the MAC address for an interface.

Also modify the `/etc/hosts` file, changing the mapping for the host from `127.0.0.1` to whatever it is now defined as in the `/etc/network/interfaces` file. Then:

```
tristram@frog:~$ sudo reboot
```

Next, we need to begin the process of installing some required packages. To begin with, update the package index, and then upgrade the current installation:

```
tristram@frog:~$ sudo apt-get update
```

```
...
```

```
tristram@frog:~$ sudo apt-get upgrade
```

```
...
```

```
Do you want to continue [Y/n] Y
```

```
...
```

At this point the basic system should be relatively up to date. To make life a lot easier, I suggest you now install *openssh* to allow command line access to the guest from remote machines, without needing to use the RDP server and the *rdesktop* package or equivalent.

```
tristram@frog:~$ sudo apt-get install openssh-server
```

If that has worked, you should be able to open an ssh connection to the guest either from itself (as localhost) or from another machine:

```
tcsh% ssh frog
```

Assuming that is all okay, you should be able to logout from the rdesktop console session, and then close the rdesktop window. The guest will still be running, and you can talk to it from your usual desktop machine via ssh. Remember to shut the machine down nicely when you are done with it. One of the nice things about Ubuntu is the speed with which it boots, and shuts down. My virtual machines are typically booted up in less than thirty seconds, and shut down much faster than that. Compared with the other flavours of Linux I have used, this is very good, and I find it makes the whole process of starting up a machine to start up MATLAB much more bearable.

To continue, we need to add a number of packages. The following works for me.

```
tristram@frog:~$ sudo apt-get install ia32-libs
tristram@frog:~$ sudo apt-get install nfs-common
tristram@frog:~$ sudo apt-get install openjdk-6-jdk
tristram@frog:~$ sudo apt-get install xauth
tristram@frog:~$ sudo apt-get install xserver-xorg
tristram@frog:~$ sudo apt-get install libxp6
tristram@frog:~$ sudo apt-get install nvidia-current
tristram@frog:~$ sudo apt-get install libglu1-mesa-dev
tristram@frog:~$ sudo apt-get install autofs
tristram@frog:~$ sudo apt-get install tcsh
tristram@frog:~$ sudo apt-get install ntp
```

If you prefer, combine all of the install commands onto a single line, separating the package names with a space.. Autofs provides support for automounting directories from a server. I usually set up an `/etc/auto.home` file like this:

```
# Home directory map for automounter
#
+auto.home
tristram chicken:/export/home0/tristram
```

Include this from the `/etc/auto.master`:

```
...
/net -hosts
/home auto.home -nobrowse
...
```

It is likely you will need to change the user and group id that Ubuntu has assigned you on installation. Look for your username in `/etc/passwd` and make sure that the line matches with the corresponding one on your nfs server.

If you are using Ubuntu 11.x there may be some additional work required to get the *NFS* mounting to behave. Ubuntu 11.04 will use *NFSv4* by default, and this relies on **idmapd** to handle the mapping of user / group names between the server and client⁶. Choices here are either to force *NFSv3* for the automounting, by using the **-fstype=nfs4** option, or (better) to correctly configure the files `/etc/nsswitch`, `/etc/idmapd.conf` and `/etc/default/nfs-common`.

Add the following to `/etc/nsswitch`:

```
...
automount: files
...
```

In `/etc/idmapd.conf`, ensure that the Domain corresponds to that used by the server. If the server is running Solaris, the domain will usually be specified in `/etc/resolv.conf`.

```
...
Domain = quantmodels.co.uk
...
```

Finally, in `/etc/default/nfs-common`:

⁶*NFSv3* does this directly with uid / gid in numeric form.

```
...  
# Do you want to start the idmapd daemon?  
NEED_IDMAPD=yes  
...
```

Ntp provides time synchronisation between machines. I tell the guest to ask someone trustworthy via `/etc/ntp.conf`:

```
# /etc/ntp.conf, configuration for ntpd; see  
ntp.conf(5) for help  
server chicken.quantmodels.co.uk  
server piglet.quantmodels.co.uk  
driftfile /var/lib/ntp/ntp.drift
```

Reliability of the hardware clock is one of the main reasons I have switched to using Ubuntu, rather than CentOS. I tried a lot of kernel settings, but never managed to get CentOS to behave correctly in this respect⁷.

7.1 Installing MATLAB

We should now be ready to install MATLAB. Either download the installation images from the MathWorks website, or install from DVD. In theory we can present the DVD from the host system to the guest using *Virtual-Box*, but I usually find it simpler to share the relevant directory from my desktop machine, and mount it via `nfs` on the guest:

⁷I have never had clock problems on bare metal installations of CentOS, only on virtual machines.

On the desktop machine:

```
tcsh% xhost + frog
```

```
tcsh% su
```

Password:

```
tcsh# share /cdrom/mathworks_r2008b
```

On the Ubuntu guest:

```
tristram@frog:~$ sudo mkdir /tmpmnt
```

```
tristram@frog:~$ sudo mount cow:/cdrom/mathworks_r2008b /tmpmnt
```

```
tristram@frog:~$ ls /tmpmnt
```

```
install inst_doc.pdf mac_install_guide.pdf update
```

```
InstallForMacOSX.app license.txt readme.txt utils
```

```
tristram@frog:~$ DISPLAY=cow:0;export DISPLAY
```

```
tristram@frog:~$ sudo /tmpmnt/install
```

After that, follow the usual MATLAB installation steps, and you should be done.